```cpp
1  #include "Enemy.h"
2  #include <iostream>
3  #include <Windows.h>
4
5  Enemy::Enemy(int x, int y, int deltaX, int deltaY)
6      : PlaceableActor(x, y, kGreenColour) // placing initial coordinates of    ⏎
         enemy
7      , m_currentMovementX(0)
8      , m_currentMovementY(0)
9      , m_directionX(0)
10     , m_directionY(0)
11     , m_movementInX(deltaX) //  The maximum distance the enemy can move in    ⏎
         the x-direction
12     , m_movementinY(deltaY) //  The maximum distance the enemy can move in    ⏎
         the y-direction
13  {
14      if (m_movementInX != 0)
15      {
16          m_directionX = 1;
17      }
18      if (m_movementinY != 0)
19      {
20          m_directionY = 1;
21      }
22  }
23
24  void Enemy::Draw()
25  {
26      HANDLE console = GetStdHandle(STD_OUTPUT_HANDLE);
27      SetConsoleTextAttribute(console, m_colour);
28      std::cout << (char)153; // prints coloured enemy.
29      SetConsoleTextAttribute(console, kRegularColour);
30  }
31
32  void Enemy::Update() // update the state of the enemy
33  {
34      if (m_movementInX != 0)
35      {
36          updateDirection(m_currentMovementX, m_directionX, m_movementInX);
37      }
38      if (m_movementinY != 0)
39      {
40          updateDirection(m_currentMovementY, m_directionY, m_movementinY);
41      }
42
43      this->SetXYPosition(m_pPosition->x + m_directionX, m_pPosition->y +      ⏎
         m_directionY);
44  }
45
46  void Enemy::updateDirection(int& current, int& direction, int& movement) //  ⏎
     responsible for handling the movement of the enemy
47  {
48      current += direction;
```

```cpp
49        if (std::abs(current) > movement) // reverse movement. if we reach the
              end we want to loop back the other way.
50        {
51            current = movement * direction;
52            direction *= -1; // change direction
53        }
54  }
55
56  //  If the absolute value of the current movement becomes greater than the
        maximum allowed movement (movement), it means the enemy has reached the end
         of its allowed movement range
```